

PATENT APPLICATION  
Docket No. 3000.2.14  
IBM No. ST9-99-115

PATENT APPLICATION  
Docket No. 3000.2.14  
IBM No. ST9-99-115

## of

**TANYA COUCH**

**VIRGINIA HUGHES**

**THERESA H. LAI**

JOSEPH D. LEA

DAVID H. OBERSTADT

for

# SYSTEM AND METHOD FOR FILTERING QUERY STATEMENTS FROM MULTIPLE PLANS AND PACKAGES ACCORDING TO USER-DEFINED FILTERS OF QUERY EXPLAIN DATA

**MADSON & METCALF, P.C.**

ATTORNEYS AT LAW  
900 GATEWAY TOWER WEST  
15 WEST SOUTH TEMPLE  
SALT LAKE CITY, UTAH 84101

## **BACKGROUND OF THE INVENTION**

The present invention relates generally to computer-implemented database systems. More specifically, the present invention relates to a system and method for filtering query statements from multiple plans and packages according to user-defined filters of query explain data.

1       **4. The Relevant Technology**

2               Databases are computerized information storage and retrieval systems. A  
3 Relational Database Management System (RDBMS) is a database system which uses  
4 relational techniques for storing and retrieving data. Relational databases are  
5 organized into tables consisting of rows (tuples) and columns of data. A database  
6 typically includes many tables, and each table includes multiple rows and columns.  
7 The tables are conventionally stored in direct access storage devices (DASD), such  
8 as magnetic or optical disk drives, for semi-permanent storage.

9               Generally, users communicate with an RDBMS using a Structured Query  
10 Language (SQL) interface. The SQL interface allows users to create, manipulate,  
11 and query a database by formulating relational operations on the tables, either  
12 interactively, in batch files, or embedded in host languages such as C and COBOL.  
13 SQL has evolved into a standard language for RDBMS software and has been  
14 adopted as such by both the American National Standards Institute (ANSI) and the  
15 International Standards Organization (ISO).

16              The SQL standard provides that each RDBMS should respond to a particular  
17 query in the same way, regardless of the underlying database. However, the  
18 method that the RDBMS actually uses to find the requested information in the  
19 database is left to the RDBMS. Typically, there is more than one method that can  
20 be used by the RDBMS to access the requested data. The RDBMS, therefore,  
21 attempts to select the method that minimizes the computer time and resources (i.e.  
22 cost) for executing the query.

23              The RDBMS determines how to execute the SQL statements. The set of steps  
24 created by the RDBMS for executing the SQL statements is commonly referred to  
25 as the "access path." In other words, the access path is a sequence of operations  
26 used by the RDBMS to obtain the data requested by the SQL query. Depending on

1 the access path, an SQL statement might search an entire table space, or,  
2 alternatively, it might use an index. The access path is the key to determining how  
3 well an SQL statement performs. The description of the access path is stored in a  
4 plan table, which typically stores the access path data for a plurality of SQL  
5 statements.

6 In addition to determining the access path, some databases estimate the cost  
7 for executing each SQL statement. The estimated costs are typically stored in a  
8 statement table (as in the case of DB2® for OS/390®) or a similar table within the  
9 database. Like the plan table, the statement table stores the estimated statement  
10 costs for a plurality of SQL statements.

11 Databases also typically include statistics for such database objects as table  
12 spaces, indexes, tables, and columns. For example, in the case of a table, the  
13 statistical data may include the number of pages that contain rows of the table, the  
14 number of rows and columns in the table, as well as various other statistical data.  
15 The statistics are typically derived from the RDBMS "catalog," which is an object  
16 that describes the entire database.

17 The above-described access path data, statement cost data, and object  
18 statistics data (referred to collectively as "query explain data") assist the user in  
19 analyzing and improving the performance of SQL statements. For example, a query  
20 with a higher-than-average statement cost might alert the user to inefficiencies in  
21 the access path. By viewing the access path data, the user can selectively make  
22 changes to the query and/or the database, such as by adding an index in order to  
23 avoid a table space scan. The object statistics data similarly assists the user by  
24 describing the structure and organization of the database.

25 Unfortunately, analyzing SQL query performance is currently too time-  
26 consuming and complex for the average user. Typical systems include hundreds

1 or thousands of query statements. Identifying the one or more statements that need  
2 to be improved can be a difficult task. In conventional systems, the user must  
3 manually locate the relevant query explain data in a plurality of tables, e.g. the plan  
4 table, the statement table, the function table, and the catalog tables, which can be  
5 tedious and time-consuming.

6 Moreover, each of the tables typically stores information corresponding to  
7 many different statements and objects. Consequently, the tables are often very  
8 large, making it difficult to locate the desired data. Likewise, the tables are often  
9 cryptic and hard to understand, even for database experts. For example, the plan  
10 table typically includes sixty or more columns and hundreds or thousands of rows.  
11 The access path data is stored in a tabular format, which, although easily  
12 understood by the RDBMS, is often too complicated to be effectively analyzed.

13 Furthermore, the query statements to be filtered are typically included in a  
14 number of packages and plans. A package is a collection of query statements found  
15 in a single application program. A plan is also a collection of query statements, but  
16 may include statements from one or more application programs. In large-scale  
17 database systems, the number of plans and packages is correspondingly large.  
18 Thus, a difficulty arises when trying to locate a particular package or plan for  
19 purposes of analyzing the query statements contained therein.

20 Accordingly, what is needed is a system, method, and article of manufacture  
21 for selectively filtering, reducing, or otherwise focusing a list of query statements  
22 based on a user's customized data requirements for improving SQL performance.  
23 Additionally, what is needed is a system, method, and article of manufacture for  
24 filtering query statements according to user-defined filters of query explain data.  
25 What is also needed is a system, method, and article of manufacture for generating  
26 user-defined filters of object reference data, statement cost data, and access path

1 data. Moreover, what is needed is a system, method, and article of manufacture for  
2 storing and retrieving sets of user-defined filters. In addition, what is needed is a  
3 system, method, and article of manufacture for selectively filtering, reducing, or  
4 otherwise focusing a list of packages or plans according to user-defined filtering  
5 criteria.

## SUMMARY OF THE INVENTION

The present invention solves the foregoing problems by providing a system and method for filtering query statements from multiple plans and packages according to query explain data associated therewith. Additionally, the present invention provides a system and method for filtering a plurality of groups of query statements, such as one or more plans and packages, by applying user-specified filtering criteria to associated identification data.

In one aspect of the invention, an apparatus for filtering a plurality of groups of query statements according to identification data associated therewith includes a processor for executing instructions and a memory device having thereon modules of operational data and executable code for execution by the processor. In one embodiment, the modules include at least one filtering module configured to receive one or more user-specified filtering criteria directed to a subset of the identification data and apply the filtering criteria to selectively exclude those of the plurality of groups not satisfying the filtering criteria. In one embodiment, the modules also include a group listing module configured to display to a user those of the plurality of groups not excluded by the at least one filtering module.

In another aspect of the invention, a method for filtering a plurality of groups of query statements according to identification data associated therewith includes the step of receiving one or more user-specified filtering criteria directed to a subset of the identification data. In one embodiment, the method also includes the step of applying the filtering criteria to selectively exclude those of the plurality of groups not satisfying the filtering criteria. Additionally, the method includes in one embodiment the step of displaying to a user those of the plurality of groups not excluded by the filtering criteria.

1 In yet another aspect of the invention, an article of manufacture comprises  
2 a program storage medium readable by a processor and embodying one or more  
3 instructions executable by the processor to perform the above-described method for  
4 filtering a plurality of groups of query statements according to associated  
5 identification data.

6 These and other objects, features, and advantages of the present invention  
7 will become more fully apparent from the following description and appended  
8 claims, or may be learned by the practice of the invention as set forth hereinafter.

## BRIEF DESCRIPTION OF THE DRAWINGS

These and other more detailed and specific objects and features of the present invention are more fully disclosed in the following specification, reference being had to the accompanying drawings, in which

Figure 1 is a schematic block diagram of a computer system suitable for implementing one embodiment of the invention;

Figure 2 is a schematic block diagram of a system for filtering query statements according to user-defined filters of query explain data according to one embodiment of the invention;

Figure 3 is a schematic block diagram of a query visualization module according to one embodiment of the invention;

Figure 4 is an illustration of a query statement and a portion of a plan table according to one embodiment of the invention;

Figure 5 is an illustration of a graphical representation of an access path according to one embodiment of the invention;

Figure 6 is a schematic block diagram of a filter module according to one embodiment of the invention;

Figure 7 is an illustration of the data flow within a filter module according to one embodiment of the invention;

Figure 8 is a schematic flow chart of a method for filtering query statements according to user-defined filters of query explain data;

Figure 9 is an illustration of an interactive display provided by a package selection module according to one embodiment of the invention;

Figure 10 is an illustration of an interactive display provided by a plan selection module according to one embodiment of the invention;

1 Figure 11 is an illustration of an unfiltered list of query statements according  
2 to one embodiment of the invention;

3 Figure 12 is an illustration of an interactive display provided by an object  
4 filter generation module according to one embodiment of the invention;

5 Figure 13 is an illustration of a filtered list of query statements according to  
6 one embodiment of the invention;

7 Figure 14 is an illustration of an interactive display provided by a cost filter  
8 generation module according to one embodiment of the invention;

9 Figure 15 is an illustration of a filtered list of query statements according to  
10 one embodiment of the invention;

11 Figure 16 is an illustration of an interactive display provided by an access  
12 path filter generation module according to one embodiment of the invention;

13 Figure 17 is an illustration of an interactive display provided by an access  
14 path filter generation module according to one embodiment of the invention;

15 Figure 18 is an illustration of a filtered list of query statements according to  
16 one embodiment of the invention.

17 Figure 19 is a schematic block diagram of a statement selection module  
18 according to one embodiment of the invention;

19 Figure 20 is a schematic flow chart illustrating a method for filtering groups  
20 of query statements according to one embodiment of the invention;

21 Figure 21 is an illustration of an interactive display provided by a package  
22 selection module according to one embodiment of the invention;

23 Figure 22 is an illustration of an interactive display provided by a package  
24 selection module according to one embodiment of the invention;

25 Figure 23 is an illustration of a filtered list of packages provided by a  
26 package listing module according to one embodiment of the invention;

1 Figure 24 is an illustration of an interactive display provided by a package  
2 selection module according to one embodiment of the invention;

3 Figure 25 is an illustration of an interactive display provided by a plan  
4 selection module according to one embodiment of the invention;

5 Figure 26 is an illustration of an interactive display provided by a plan  
6 selection module according to one embodiment of the invention;

7 Figure 27 is an illustration of a filtered list of plans provided by a plan listing  
8 module according to one embodiment of the invention; and

9 Figure 28 is an illustration of an interactive display provided by a plan  
10 selection module according to one embodiment of the invention.

1                    **DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS**

2                    The presently preferred embodiments of the present invention will be best  
3 understood by reference to the drawings, wherein like parts are designated by like  
4 numerals throughout. It will be readily understood that the components of the  
5 present invention, as generally described and illustrated in the figures herein, may  
6 be arranged and designed in a wide variety of different configurations. Thus, the  
7 following more detailed description of the embodiments of the apparatus, system,  
8 and method of the present invention, as represented in the Figures, is not intended  
9 to limit the scope of the invention, as claimed, but is merely representative of  
10 presently preferred embodiments of the invention.

11                  The Figures include schematic block diagrams and flow chart diagrams  
12 which illustrate in more detail the preferred embodiments of the present invention.  
13 The schematic block diagrams illustrate certain embodiments of modules for  
14 performing various functions of the present invention. In general, the represented  
15 modules include therein executable and operational data for operation within a  
16 computer system of Figure 1 in accordance with the present invention.

17                  As used herein, the term executable data, or merely an "executable," is  
18 intended to include any type of computer instructions and computer executable  
19 code that may be located within a memory device and/or transmitted as electronic  
20 signals over a system bus or network. An identified module of executable code  
21 may, for instance, comprise one or more physical or logical blocks of computer  
22 instructions which may, for instance, be organized as an object, procedure, or  
23 function. Nevertheless, the executables of an identified module need not be located  
24 together, but may comprise disparate instructions stored in different locations  
25 which together comprise the module and achieve the purpose stated for the module.  
26 Indeed, an executable may be a single instruction, or many instructions, and may

1 even be distributed over several different code segments, among different  
2 programs, and across several memory devices.

3 Similarly, operational data may be identified and illustrated herein within  
4 modules, and may be embodied in any suitable form and organized within any  
5 suitable type of data structure. The operational data may be collected as a single  
6 data set, or may be distributed over different locations including over different  
7 storage devices, and may at least partially exist merely as electronic signals on a  
8 system bus or network.

9 *Fig. 1* Figure 1 is a schematic block diagram that illustrates a computer system 10  
10 in which executable and operational data, operating in accordance with the present  
11 invention, may be hosted on one or more computer stations 12 in a network 14. The  
12 network 14 may comprise a wide area network (WAN) or local area network (LAN)  
13 and may also comprise an interconnected system of networks, one particular  
14 example of which is the Internet and the World Wide Web supported on the  
15 Internet.

16 A typical computer station 12 may include a processor or CPU 16. The CPU  
17 16 may be operably connected to one or more memory devices 18. The memory  
18 devices 18 are depicted as including a non-volatile storage device 20 such as a hard  
19 disk drive or CD-ROM drive, a read-only memory (ROM) 22, and a random access  
20 volatile memory (RAM) 24. Preferably, the computer station 12 operates under the  
21 control of an operating system (OS) 25, such as MVS®, OS/390®, AIX®, OS/2®,  
22 WINDOWS NT®, WINDOWS®, UNIX®, and the like.

23 The computer station 12 or system 10 in general may also include one or  
24 more input devices 26, such as a mouse or keyboard, for receiving inputs from a  
25 user or from another device. Similarly, one or more output devices 28, such as a  
26 monitor or printer, may be provided within or be accessible from the computer

1 system 10. A network port such as a network interface card 30 may be provided for  
2 connecting to outside devices through the network 14. In the case where the  
3 network 14 is remote from the computer station, the network interface card 30 may  
4 comprise a modem, and may connect to the network 14 through a local access line  
5 such as a telephone line.

6 Within any given station 12, a system bus 32 may operably interconnect the  
7 CPU 16, the memory devices 18, the input devices 26, the output devices 28 the  
8 network card 30, and one or more additional ports 34. The system bus 32 and a  
9 network backbone 36 may be regarded as data carriers. As such, the system bus 32  
10 and the network backbone 36 may be embodied in numerous configurations. For  
11 instance, wire, fiber optic line, wireless electromagnetic communications by visible  
12 light, infrared, and radio frequencies may be implemented as appropriate.

13 In general, the network 14 may comprise a single local area network (LAN),  
14 a wide area network (WAN), several adjoining networks, an intranet, or as in the  
15 manner depicted, a system of interconnected networks such as the Internet 40. The  
16 individual stations 12 communicate with each other over the backbone 36 and/or  
17 over the Internet 40 with varying degrees and types of communication capabilities  
18 and logic capability. The individual stations 12 may include a mainframe computer  
19 on which the modules of the present invention may be hosted.

20 Different communication protocols, e.g., ISO/OSI, IPX, TCP/IP, may be  
21 used on the network, but in the case of the Internet, a single, layered  
22 communications protocol (TCP/IP) generally enables communications between the  
23 differing networks 14 and stations 12. Thus, a communication link may exist, in  
24 general, between any of the stations 12.

25 The stations 12 connected on the network 14 may comprise application  
26 servers 42, and/or other resources or peripherals 44, such as printers and scanners.

1 Other networks may be in communication with the network 14 through a router 38  
2 and/or over the Internet 40.

3 Referring now to Figure 2, a schematic block diagram of one embodiment  
4 of the invention includes first and second stations 12A, 12B. The first station 12A  
5 is preferably a workstation-class computer, such as an PC workstation, available  
6 from IBM Corporation. The second station 12B is preferably an IBM mainframe  
7 computer operating under MVS® or OS/390®. In one embodiment, the stations 12A,  
8 12B are coupled via a network 14 using a distributed remote data architecture  
9 (DRDA). Those skilled in the art, however, will recognize that the invention may  
10 be implemented using a variety of computing platforms and/or network  
11 architectures.

12 In one embodiment, the first station 12A includes a query visualization  
13 module 50, which is a tool that assists a user in visualizing or otherwise  
14 understanding query explain data for one or more queries to be executed. As  
15 described above, the query explain data includes, in one embodiment, the access  
16 path data, the statement cost data, and the object statistics data used in analyzing  
17 and improving the performance of SQL statements.

18 The second station 12B preferably stores a database 52, as well as an RDBMS  
19 54 for managing the database 52, such as DB2® for OS/390®, available from IBM. As  
20 used herein, the term "database" may generically refer to a combination of the  
21 RDBMS 54 and the database 52. In one embodiment, the query visualization  
22 module 50 and the RDBMS 54 are linked via an interface module 56, such as DB2  
23 Connect®, also available from IBM.

24 Referring now to Figure 3, the query visualization module 50 preferably  
25 includes a plurality of modules containing executable and operational data suitable  
26 for operation within the memory devices 18 of Figure 1. Of course, the memory

1 devices 18 in which the modules of the present invention are located may also be  
2 distributed across both local and remote computer stations 12. Likewise, two or  
3 more illustrated modules may be integrated into a single module, or the function  
4 of a single module could be performed by a group of modules, without departing  
5 from the scope of the invention.

6 The principal components of the query visualization module 50 include a  
7 report creator 60, a graph generator 62, and a parameter browser 64. The above-  
8 described modules are, in one embodiment, intended to help the user to better  
9 understand the query explain data, subsystem parameters, and the like, in a variety  
10 of ways.

11 For example, the report creator 60 in one embodiment selectively prepares  
12 a report of the access path data, statement cost data, and object statistics data in an  
13 easily understood, text-based format. The user may be provided with the option of  
14 selecting one or more query statements, as well as subsets of the query explain data  
15 for the selected query statements to include in the report. The report provides the  
16 user with the requested query explain data in a centralized and readily understood  
17 format, allowing the user to efficiently analyze and improve SQL query  
18 performance. The report creator 60 is more fully described in co-pending U.S.  
19 application Serial No. \_\_\_\_\_, filed \_\_\_\_\_, using Express Mail Label  
20 EL409135377US, for "System and Method for Selectively Preparing Customized  
21 Reports of Query Explain Data," which is commonly assigned and is incorporated  
22 herein by reference.

23 A second principal component of the query visualization module 50 in one  
24 embodiment is the graph generator 62, which prepares a graphical representation  
25 of the access path of a query statement. The graph generator 62 is more fully  
26 described in co-pending application Serial No. 08/949,636, filed October 14, 1997,

1 for "Interpreting Data Using a Graphical User Interface," which is incorporated  
2 herein by reference. Figure 4 illustrates an exemplary SQL query statement,  
3 processed by an RDBMS 54 to generate access path data and stored within a plan  
4 table 74 according to one embodiment of the present invention.

5 As shown in Figure 5, the graph generator 62 in one embodiment uses the  
6 plan table 74 to generate a graphical representation of the access path. Preferably,  
7 access path steps of an SQL statement are graphically represented as nodes within  
8 a tree-like structure. Tables, indexes, and operations are graphically represented  
9 with unique symbols that indicate the item being represented. For example,  
10 rectangles represent tables, triangles represent indexes, and octagons represent  
11 operations such as table space scans, index scans, joins, etc. The graphical  
12 representation shows the relationship between the database objects and the  
13 operations. When the user selects a node of the graphical representation, detailed  
14 information related to the selected node is displayed on the right side of the display.

15 Referring again to Figure 3, a third principal component of the query  
16 visualization module 50 in one embodiment is the parameter browser 64.  
17 Preferably, the parameter browser 64 allows a user to selectively view the  
18 subsystem parameters, for example, DSNZPARM and DSNHDECP values, used by  
19 a subsystem, as well as the install panel names and fields. Access to subsystem  
20 parameters is useful in analyzing the performance of query statements. Like the  
21 graph generator 62, the parameter browser 64 is more fully described in co-pending  
22 application Serial No. 08/949,636, filed October 14, 1997, for "Interpreting Data  
23 Using a Graphical User Interface."

24 The report creator 60, graph generator 62, and parameter browser 64 are  
25 each preferably coupled to a graphical user interface (GUI) module 66. Preferably,  
26 the GUI module 66 is operably coupled to the input and output devices 26, 28 to

1 allow the user to interact with the report creator 60, graph generator 62, and  
2 parameter browser 64.

3 The query visualization module 50 in the depicted embodiment also  
4 includes an explain module 68, which invokes a corresponding explain function 70  
5 in the RDBMS 54. When invoked, the explain function 70 causes the RDBMS 54 to  
6 generate function data, access path data, and the statement cost data (referred to  
7 herein as "explain data") for a specified query statement.

8 Explainable query statements include SELECT (except for SELECT INTO)  
9 and INSERT statements, and the searched form of UPDATE and DELETE  
10 statements. The explain module 68 is used to invoke an explain function 70 in  
11 which the RDBMS 54 immediately generates explain data for a specific SQL  
12 statement. This feature is useful for interactively testing specified SQL statements.  
13 Alternatively, the RDBMS 54 generates the explain data at bind time in the context  
14 of an application or package upon encountering an EXPLAIN(YES) open of the  
15 BIND command.

16 In one embodiment, the query visualization module 50 includes a plurality  
17 of querying modules for querying various tables in the database 52. For example,  
18 a plan table querying module 72 queries a plan table 74 to obtain access path data.  
19 Likewise, a statement table querying module 76 queries a statement table 78 to  
20 obtain statement cost data. A function table querying module 80 queries a function  
21 table 82 to obtain data concerning user-defined functions. The plan table 74, the  
22 statement table 78, and the function table 82 are collectively referred to herein as  
23 "explain tables." Finally, a catalog querying module 86 queries the RDBMS catalog  
24 88 to obtain object statistics for one or more database objects contained within in a  
25 plurality of user tables 90.  
26

1           Although the querying function is implemented herein by four separate  
2 modules, those skilled in the art will recognize that the described functionality may  
3 be implemented by fewer modules. Additionally, in one embodiment, the above-  
4 described modules use the interface module 56 when communicating with the  
5 RDBMS 54 and database 52.

6           Preferably, the query visualization module 50 also includes a filter module  
7 92, which is described in greater detail below. In one embodiment, the filter module  
8 92 allows a user to filter a list of explainable query statements according to various  
9 user-selected criteria, including statement costs, references to particular database  
10 objects, and the inclusion of particular steps in the access paths of the statements.  
11 Moreover, in one embodiment, the user may assign a name to a set of filtering  
12 criteria and save the named set in a filter storage 94. This allows the user to easily  
13 retrieve a specified set of filtering criteria such that only the explainable query  
14 statements satisfying the criteria are listed.

15           In one embodiment, the filter module 92 also allows a user to filter a set of  
16 plans and packages according to various identification data, such as plan and  
17 package names, collection identifiers, version identifiers, and the like. The filter  
18 module 92 is described in greater detail below with respect to Figure 6.

19           The query visualization module 50 also preferably includes a cache module  
20 96, which caches portions of the above-described tables in a cache storage 98. For  
21 example, when the plan table querying module 72 retrieves access path data from  
22 the plan table 74, the access path data is preferably stored, and future accesses to the  
23 same data will be retrieved from the cache storage 98. Various methods may be  
24 employed for managing data in the cache storage 98, such as automatically deleting  
25 a percentage of the cached data when the amount of the data exceeds a pre-defined  
26 threshold.

1 Referring now to Figure 6, the filter module 92 preferably includes a  
2 plurality of modules containing executable and operational data suitable for  
3 operation within the memory devices 18 of Figure 1. In one embodiment, the  
4 memory devices 18 in which the modules of the present invention are located may  
5 also be distributed across both local and remote computer stations 12. Likewise,  
6 two or more illustrated modules may be integrated into a single module without  
7 departing from the scope of the invention, and additional modules could be utilized  
8 to perform the same functions.

9 In one embodiment, the filter module 92 includes a statement selection  
10 module 100, which allows a user to select an initial set of query statements to be  
11 filtered. Briefly, a package comprises one or more query statements from a common  
12 application. Likewise, a plan is an assemblage of query statements, but may  
13 correspond to more than one application. The user may preferably select one or  
14 more plans or packages to create the initial set of query statements.

15 As described above, the initial set of query statements may be very large,  
16 particularly for a company-wide database. Consequently, it may be difficult for a  
17 user to locate a particular query statement, when, for example, the user wishes to  
18 select the statement for graphing, for SQL text display, for generation of a report  
19 containing the statement, or the like.

20 To alleviate this problem, the filter module 92 preferably includes a filter  
21 generation module 102. In one embodiment, the filter generation module 102 allows  
22 the user to create one or more filters for selectively reducing a list of query  
23 statements based on subsets of the query explain data previously generated for the  
24 statements by the RDBMS 54. In the depicted embodiments, each filter includes  
25 user-specified filtering criteria that are applied to the corresponding subset of the  
26

1 query explain data to selectively exclude from the initial set the query statements  
2 not satisfying the filtering criteria.

3 The filter generation module 102 preferably includes an object filter  
4 generator 104 for assisting a user in generating a user-defined object filter 106. In  
5 one embodiment, the object filter 106 includes filtering criteria for selectively  
6 excluding query statements from the initial set based on whether the query  
7 statements either reference a specified database object (such as a table, index, or the  
8 like) or reference an object created by a specified person.

9 Similarly, the filter generation module 102 also includes, in one embodiment,  
10 a cost filter generator 108 for assisting a user in generating a user-defined cost filter  
11 110. The cost filter 110 preferably includes filtering criteria for selectively excluding  
12 query statements from the initial set based on how the statement cost data for each  
13 query statement compares with a specified threshold cost.

14 Likewise, the filter generation module 102 preferably includes an access path  
15 filter generator 112 for assisting a user in generating a user-defined access path filter  
16 114. The access path filter 114 preferably includes filtering criteria for selectively  
17 excluding query statements from the initial set based on whether the statements  
18 include specified execution steps in their access paths. Other types of filter  
19 generators may also be added as needed.

20 To apply the foregoing filters, the filter module 92 preferably includes a  
21 statement filtering module 116. In one embodiment, the statement filtering module  
22 116 applies the filtering criteria of the filters to subsets of the query explain data for  
23 the initial set of query statements.

24 In the depicted embodiment, the statement filtering module 116 includes an  
25 object criteria module 118 for applying the filtering criteria of an object filter 106.  
26 Preferably, the object criteria module 118 communicates with the plan table

1 querying module 72 to obtain the access path data for each query statement from  
2 the plan table 74. From the access path data, the object criteria module 118 may  
3 determine whether each statement references a database object specified in the  
4 filtering criteria, such as a table, index, or the like.

5         Additionally, the statement filtering module 116 preferably includes a cost  
6 criteria module 120 for applying the filtering criteria of a cost filter 110. In one  
7 embodiment, the cost criteria module 120 communicates with the statement table  
8 querying module 76 to obtain the statement cost data for each query statement from  
9 the statement table 78. From the statement cost data, the cost criteria module 120  
10 may determine whether each statement satisfies the filtering criteria of the cost filter  
11 110.

12         In one embodiment, the statement filtering module 116 also includes an  
13 access path criteria module 122 for applying the filtering criteria of an access path  
14 filter 114. Preferably, the access path criteria module 122 communicates with the  
15 plan table querying module 72 to obtain the access path data for each query  
16 statement from the plan table 74. From the access path data, the access path criteria  
17 module 122 may determine whether the access path of each statement includes  
18 execution steps specified in the filtering criteria of the access path filter 114.

19         In one embodiment, the filter module 92 also includes a statement listing  
20 module 124. The statement listing module 124 preferably displays the filtered set  
21 of query statements to the user, additionally allowing the user to select one or more  
22 of the query statements for purposes of modifying the statements, including the  
23 statements in a report, explaining the query statements, or the like.

24         In one embodiment, the filter module 92 also includes a filter storage  
25 module 126. After the user has generated one or more filters, the filter storage  
26 module 126 preferably allows the user to store the filters as a set 130 in the filter

1 storage 94. In one embodiment, the user may assign a name to a set 130 of filters in  
2 the filter storage 94. Later, the user may retrieve and use the stored filters without  
3 having to recreate them, as described in greater detail below.

4 Figure 7 is an illustration of the data flow within the filter module 92  
5 according to one embodiment of the invention. As described above, the RDBMS 54  
6 generates query explain data for a plurality of query statements. Using the  
7 statement selection module 100, the user may select one or more plans and/or  
8 packages, which define an initial set of query statements to be filtered.

9 A user may, using the filter generation module 102, generate one or more  
10 filters, such as an object filter 106, a cost filter 110, and an access path filter 114.  
11 Optionally, the filters may be stored in the filter storage 94 by means of the filter  
12 storage module 126. Later, a user may retrieve the filters from the filter storage 94.  
13 In one embodiment, the filters may include the user's selection of the plans and/or  
14 packages.

15 In one embodiment, the statement filtering module 116 accepts as input the  
16 initial set of query statements, the query explain data for the statements, and one or  
17 more user-defined filters, either generated by the filter generation module 102 or  
18 retrieved from the filter storage 94. Using the techniques described above, the  
19 statement filtering module 116 applies the filtering criteria of each user-defined filter  
20 to selectively exclude from the initial set the query statements not satisfying the  
21 filtering criteria. Preferably, the statement filtering module 116 provides the filtered  
22 query statements to the statement listing module 124, which displays a list of the  
23 filtered query statements to the user.

24 Referring now to Figure 8, a schematic flow chart illustrates a method of  
25 filtering query statements according to one embodiment of the invention. The  
26 method begins by determining 132 whether to create new filters or to retrieve a set

1 130 of stored filters. In the depicted embodiment, a user may create new filters by  
2 selecting a "new" button 146 or a similar control. Alternatively, a user may retrieve  
3 a set 130 of filters from the filter storage 94 by selecting the name of a set 130 from  
4 a pull-down menu 148 (of Figure 10), which lists the names of the filter sets 130  
5 stored in the filter storage 94.

6 If, in step 132 of Figure 8, the user elects to create new filters, the method  
7 continues by receiving 134 a user's selection of one or more plans and/or packages.  
8 As noted above, a package corresponds to one or more query statements in a  
9 common application. Likewise, a plan is an assemblage of query statements, but  
10 may correspond to more than one application.

11 Figures 9 and 10 are illustrations of interactive displays, generated by the  
12 statement selection module 100 for receiving a user's selection of one or more plans  
13 and/or packages. Figure 9 illustrates an interactive display for selecting packages  
14 according to one embodiment of the invention. Preferably, the statement selection  
15 module 100 provides an "add" button 134, or a similar control, through which the  
16 user may select to add one or more packages to a selected package list 136. In one  
17 embodiment, the user may add packages by name or by performing a search of the  
18 packages based on such criteria as package owner, package name, collection ID, and  
19 version. This process is described in greater detail below with respect to Figures 19-  
20 28.

21 In one embodiment, the selection module 100 employs a control such as a  
22 "change" button 138 for changing one or more of the selected packages in the  
23 selected package list 136. In addition, the selection module 100 may include a  
24 "remove" button 140 or similar control for removing one or more of the selected  
25 packages.

26 Figure 10 illustrates an interactive display for selecting plans according to

1 one embodiment of the invention. Preferably, the selection module 100 provides a  
2 control such as an "add" button 134 for adding one or more plans to a selected plan  
3 list 142. As noted above, a plan is similar to a package but may include query  
4 statements for more than one application.

5 After the user's selection of plans and/or packages is received, the method  
6 continues by receiving 150 user-specified filtering criteria, if any, for an object filter  
7 106. It should be recognized that a user is not required in one embodiment to create  
8 a complete set of filters or, indeed, any filters. Thus, if the user selects an "OK"  
9 button 152 (of Figure 10) or the like without creating any filters, the statement listing  
10 module 124 (of Figure 7) preferably lists all of the query statements in the selected  
11 plans and/or packages. Figure 11 illustrates an unfiltered listing of the plurality of  
12 query statements for the selected plans of Figure 10. If the user does not define an  
13 object filter 106, the method simply proceeds with step 168.

14 Figure 12 is an illustration of an interactive display provided by the object  
15 filter generator 104 for receiving user-specified filtering criteria of an object filter  
16 106. In the depicted embodiment, the interactive display for the object filter  
17 generator 104 is shared with the cost filter generator 108. However, in alternative  
18 embodiments, separate interactive displays are provided. Preferably, the interactive  
19 display for the cost filter generator 108 is selectively displayed by means of a  
20 suitable control such as a "Cost / object filters" tab 153.

21 In one embodiment, the interactive display includes a set of customizable  
22 relational expressions 154 for specifying filtering criteria of an object filter 106. Each  
23 relational expression preferably represents an individual filtering criterion that is  
24 applied to the object reference data for each query statement. Although the  
25 following description illustrates a preferred user interface, a variety of other user  
26 interfaces may be employed within the scope of the invention.

1 In the depicted embodiment, each relational expression 154 includes three  
2 fields: a name field 156, an operator field 158, and a value field 160. The name field  
3 156 preferably represents a type of database object referenced by a query statement,  
4 such as a table or an index. In addition, the name field 156 may represent an object  
5 creator, i.e. a person who created one or more referenced objects.

6 The operator field 158 is preferably selectable from a number of relational  
7 operators, such as "equal to," "not equal to," "greater than," "less than," "in" or  
8 "like." In one embodiment, the operator field 158 may be selected using a pull  
9 down menu or similar mechanism.

10 Preferably, the value field 160 stores the name of a specific object or a name  
11 of a specific object creator. In one embodiment, the value field 160 is specified using  
12 a text entry field or like mechanism.

13 For example, as shown in Figure 12, the expression 154A includes "Table  
14 names" in the name field 156, "=" in the operator field 158, and "TBDB0101" in the  
15 value field 160. Thus, in one embodiment, the resulting relational expression,  
16 "Table names = TBDB0101," specifies that only the query statements referencing the  
17 "TBDB0101" table will pass through the object filter 106; all other statements will be  
18 excluded. For example, as displayed by the statement listing module 124 in Figure  
19 13, each of the filtered statements references the "TBDB0101" table.

20 In one embodiment, a filter may include a plurality of relational expressions  
21 combined with boolean operators. In the illustrated embodiment of Figure 12, the  
22 relational expressions 154 may be combined using a logical (inclusive) "OR" by  
23 selecting the "Meet any filter condition" option 162. Alternatively, the relational  
24 expressions 154 may be combined using a logical "AND" by selecting the "Meet all  
25 filter conditions" option 164. Other boolean operations may also be provided, such  
26 as NOT, exclusive-OR, and the like.

1 Referring again to Figure 8, after the filtering criteria for the object filter 106  
2 is received, the method continues by receiving 168 filtering criteria, if any, for a cost  
3 filter 110. Figure 14 is an illustration of an interactive display provided by the cost  
4 filter generator 108 for receiving user-specified filtering criteria of a cost filter 110.  
5 Preferably, the interactive display for the cost filter generator 108 may be selectively  
6 displayed by means of the "Cost / object filters" tab 153.

7 In one embodiment, the interactive display includes a set of customizable  
8 relational expressions 170 for specifying the filtering criteria for a cost filter 110.  
9 Each relational expression 170 preferably represents an individual filtering criterion  
10 that is applied to the statement cost data for each query statement.

11 The relational expressions 170 for the cost filter 110 are similar in form to  
12 those of the object filter 106. However, the name field 156 preferably represents cost  
13 units for measuring the cost of executing the query statement. In one embodiment,  
14 the cost units are either milliseconds or service units. Service units are a measure  
15 of a computer's processing resources needed to execute the query statement. The  
16 value field 160 preferably stores a threshold cost to be compared with the statement  
17 cost of each query statement.

18 For example, as shown in Figure 14, the expression 170A includes "Cost in  
19 milliseconds" in the name field 156, ">" in the operator field 158, and "1000" in the  
20 value field 160. Thus, the relational expression, "Cost in milliseconds > 1000"  
21 specifies that only those query statements with an estimated cost greater than 1000  
22 milliseconds will pass through the cost filter 110; all other statements will be  
23 excluded. For example, as displayed by the statement listing module 124 in Figure  
24 15, each of the filtered statements has a cost greater than 1000 milliseconds.

25 Referring again to Figure 8, after the filtering criteria for the cost filter 110  
26 is received, the method continues by receiving 172 filtering criteria, if any, for an

1 access path filter 114. Figure 16 is an illustration of an interactive display provided  
2 by the access path filter generator 112 for receiving user-specified filtering criteria  
3 of an access path filter 114. Preferably, the interactive display for the access path  
4 filter generator 112 is selectively displayed by means of an "Access path filters" tab  
5 174.

6 In one embodiment, the access path filter generator 112 displays a plurality  
7 of customizable boolean expressions 176. Preferably, each of the boolean  
8 expressions 176 represents access path steps. For example, as depicted in Figure 16,  
9 the access path steps may include sorts, table space scans, and the like.

10 In one embodiment, each of the boolean expressions 176 include a name  
11 field 156 and a value field 160. The name field 156 preferably includes a type of  
12 access path step. The value field 160 is preferably a truth value, i.e. true or false. In  
13 one embodiment, the value field 160 is selectively changeable by clicking on the  
14 associated field with a mouse or other pointing device.

15 Query statements are preferably allowed to pass through the access path  
16 filter 114 if the statements include any of the access path steps having a value field  
17 160 of true. In one embodiment, the individual boolean expressions 176 are  
18 combined using a logical OR. In alternative embodiments, the expressions 176 may  
19 be combined using other boolean operators, such as AND.

20 In one embodiment, the access path filter generator 112 provides a "Set to  
21 all true" button 177 for conveniently setting all of the value fields 160 to true.  
22 Likewise, the access path filter generator 112 provides a "Set to all false" button 178  
23 for setting all of the value fields 160 to false.

24 As illustrated in Figure 17, a user may desire to see a list of query statements  
25 including sorts in the statements' access paths. This may be accomplished in one  
26 embodiment by setting all of the value fields 160 to false using the "Set all to false"

1 button 178 and changing the "Sorts" value field 160 to true. Accordingly, when the  
2 access path filter 114 is applied, only the statements having sorts are displayed. For  
3 example, as displayed by the statement listing module 124 in Figure 18, all of the  
4 listed query statements include sorts in the statements' access paths.

5 After the filtering criteria for the access path filter 114 is received, the  
6 method continues by creating 180 the filters corresponding to the user-defined  
7 filtering criteria. In one embodiment, the filters are data structures including  
8 representations of the relational expressions 154, 170 and boolean expressions 176.  
9 In addition, each filter may include, in one embodiment, the selection of plans  
10 and/or packages made by the user in step 144. The type of data structure is not  
11 crucial to the invention. However, it is preferred that the statement filtering module  
12 116 is capable of interpreting the data structures to obtain the relational expressions  
13 154, 170 and boolean expressions 176, for the filtering criteria, as well as the  
14 plan/package selection.

15 After the filters are created, the method continues by determining 182  
16 whether to save the user-specified filter(s) as a set 130 in the filter storage 94. The  
17 user may elect (or may have previously elected) to give a name to a set 130 of filters  
18 and store the set 130 by means of the "Save" or "Save as" buttons 184, 186.

19 If the user selects one of the foregoing buttons 184, 186, the method  
20 continues with step 188 by receiving the name of the filter set 130 (if not previously  
21 provided) and storing the filter set 130 in the filter storage 94 using the specified  
22 name; otherwise, the method continues with step 192.

23 As previously noted in step 144, a user may elect to retrieve a stored filter  
24 set 130, rather than creating new filters by means of the filter generation module  
25 102. This is accomplished, in one embodiment, by selecting a set 130 from a pull  
26 down menu 148, which lists the names of the filter sets 130 stored in the filter

1 storage 94. Thus, if the user elects to retrieve a stored set in step 144, the method  
2 continues by receiving 190 a name of a user-specified filter set 130 and retrieving the  
3 set 130 from the filter storage 94.

4 In one embodiment, the user may elect to modify the retrieved filter 190  
5 using the techniques described in steps 150, 168, and 172. As noted above, the  
6 filters may include, in one embodiment, the selection of plans and/or packages  
7 made by the user in step 144. However, in an alternative embodiment, the user may  
8 proceed to specify a new or different set of plans and/or packages as described in  
9 step 132.

10 After the completion of either steps 182, 188, or step 190, one or more filters  
11 will have been generated and/or retrieved. Preferably, the user requests a list of  
12 filtered statements by selecting the "OK" button 152. Thereafter, the method  
13 continues by applying 192 the object filter 106, if any, to the query statements. After  
14 the object filter 106 is applied, the method continues by applying 194 the cost and  
15 access path filters 110, 114 if any were defined.

16 When the filtering process is complete, the statement listing module 124  
17 displays a list of the filtered query statements to the user. As shown in Figure 13,  
18 the statement listing may include such information as a statement number, a  
19 package name, a cost in milliseconds, a cost in service units, the SQL text of the  
20 query statement, a collection ID, a plan name, and an owner name.

21 Additionally, in one embodiment, the statement listing module 124 allows  
22 the user to select a filtered statement for graphing, for SQL text display, for  
23 generation of a report containing the statement, or the like. Figures 13, 15, and 18  
24 illustrate an interactive display generated by the statement listing module 124 for  
25 listing the filtered query statements according to one embodiment of the invention.

26

1 Referring now to Figure 19, the statement selection module 100 preferably  
2 includes a plurality of modules containing executable and operational data suitable  
3 for operation within the memory devices 18 of Figure 1. Of course, the memory  
4 devices 18 in which the modules of the present invention are located may also be  
5 distributed across both local and remote computer stations 12. Likewise, two or  
6 more illustrated modules may be integrated into a single module, or the  
7 functionality of a single module may be implemented using a plurality of modules,  
8 without departing from the scope of the invention.

9 In one embodiment, the statement selection module 100 includes a package  
10 selection module 200 and a plan selection module 202 for assisting a user in  
11 selecting one or more packages and plans, respectively. As previously explained,  
12 packages and plans correspond to groups of query statements associated with  
13 application programs that query the database 52. Thus, by selectively designating  
14 one or more packages and plans, the user may define the initial set of query  
15 statements to be received as input to the statement filtering module 116 of Figure  
16 6.

17 In typical large-scale database systems, the number of packages and plans  
18 is correspondingly large, and finding a particular package or plan can be difficult.  
19 Accordingly, both the package selection module 200 and the plan selection module  
20 202 preferably include a number of filtering modules for selectively filtering an  
21 initial set of packages and plans according to user-specified filtering criteria.

22 In one embodiment, the filtering criteria for each filtering module is directed  
23 to identification data stored in the catalog 88 for each package and plan. Thus, in  
24 one embodiment, the catalog querying module 86 is used to access the  
25 plan/package identification data.  
26

1 For example, in one embodiment, the package selection module 200 includes  
2 a package owner filtering module 208 for filtering a set of packages according to  
3 package owner data for each package. Preferably, each package identifies an  
4 owner, i.e. a user ID of the creator or creators of the package. In one embodiment,  
5 the package owner filtering module 208 receives from a user an identification of one  
6 or more owners and filters the packages according to whether the packages were  
7 created by the specified owners.

8 Additionally, the package selection module 200 includes, in one  
9 embodiment, a package name filtering module 210 for filtering a set of packages  
10 according to package name data for each package. Preferably, each package has a  
11 unique name for identifying the package. In one embodiment, the package name  
12 filtering module 210 receives from a user the names of one or more packages and  
13 filters the packages according to whether the packages have the specified names.

14 In one embodiment, the package selection module 200 also includes a  
15 collection ID filtering module 212 for filtering a set of packages according to  
16 collection ID's. A collection ID is an identification of a group of packages. In one  
17 embodiment, the collection ID filtering module 212 receives from a user the ID's of  
18 one or more collections, and filters the packages according to whether the packages  
19 are members of the specified collections.

20 Additionally, in one embodiment, the package selection module 200  
21 includes a version filtering module 214 for filtering packages according to version  
22 data. In one embodiment, each package has a version identifier for identifying the  
23 version of the package. Preferably, the version filtering module 214 receives from  
24 a user the identifiers of one or more versions and filters the packages according to  
25 whether the packages include the specified version identifiers.

26

1 Preferably, the package selection module 200 also includes a package listing  
2 module 216 for displaying to a user a selectable list of the filtered packages. In one  
3 embodiment, the user may select one or more of the packages in order to select the  
4 initial set of query statements.

5 In one embodiment, the plan selection module 202 includes a similar set of  
6 filtering modules. For example, the plan selection module 202 includes a plan  
7 owner filtering module 218 for filtering plans according to plan owner data for each  
8 plan. Preferably, each plan identifies an owner, i.e. the user ID of the creator or  
9 creators of the plan. In one embodiment, the filtering module 218 receives from a  
10 user an identification of one or more owners and filters the plans according to  
11 whether the plans were created by the specified owners.

12 Additionally, the plan selection module 202 includes, in one embodiment,  
13 a plan name filtering module 220 for filtering plans according to plan name data for  
14 each plan. Preferably, each plan has a unique name for identifying the plan. In one  
15 embodiment, the plan name filtering module 220 receives from a user one or more  
16 names of plans and filters the plans according to whether the plans have the  
17 specified names.

18 Preferably, the plan selection module 202 also includes a database request  
19 module (DBRM) name filtering module 222 for filtering plans according to DBRM  
20 name data associated with each plan. In one embodiment, the DBRM name filtering  
21 module 222 receives from a user one or more DBRM names and filters the plans  
22 according to the specified DBRM names.

23 Preferably, the plan selection module 202 also includes a plan listing module  
24 224 for displaying to a user a selectable list of the filtered plans. In one embodiment,  
25 the user may select one or more of the plans in order to select the initial set of query  
26 statements.

1 Referring now to Figure 20, a schematic flow chart illustrates a method for  
2 filtering a set of plans and packages according to one embodiment of the invention.  
3 The method begins by receiving 230 user-specified package filtering criteria for  
4 filtering an initial set of packages.

5 Figure 21 illustrates an interactive display generated by the statement  
6 selection module 100, and more particularly, by the package selection module 200,  
7 according to one embodiment of the invention. As previously described, the  
8 statement selection module 100 preferably generates an "add" button 134 or other  
9 suitable control, which the user may select to add one or more packages to the  
10 selected package list 136.

11 As illustrated in Figure 22, when the "add" button 134 is selected, a set of  
12 customizable package filtering criteria is preferably displayed. In one embodiment,  
13 the package filtering criteria include a number of relational expressions 250.  
14 Preferably, a relational expression 250 is provided for each filtering module,  
15 including the package owner filtering module 208, the package name filtering  
16 module 210, the collection ID filtering module 212, and the version filtering module  
17 214.

18 For example, in the depicted embodiment, the package owner filtering  
19 module 208 corresponds to the relational expression 250A. Preferably, the  
20 expression 250A has the form:

21 "Package owner:" OPERATOR VALUE

22 where the OPERATOR field is selectable from a number of relational operators,  
23 such as "equal to," "not equal to," "greater than," "less than," "in," or "like," and  
24 the VALUE field stores the identifiers of one or more package owners. In one  
25 embodiment, the OPERATOR field may be selected using a pull down menu or the  
26

1 like, while the VALUE field may be supplied using a text entry box or other suitable  
2 mechanism.

3 The completed expression 250A, i.e. "Package owner: IN ADMF001,  
4 USER001, SYSADM," is preferably interpreted by the package owner filtering  
5 module 208 as an instruction to provide to the package listing module 216 only  
6 those packages owned by "ADMF001," "USER001," and "SYSADM."

7 The user may similarly select filters corresponding to package names  
8 (expression 250B), collection identifiers (expression 250C), and versions (expression  
9 250D). In one embodiment, the package name filtering module 210 corresponds to  
10 the relational expression 250B, which has a similar format to expression 250A.  
11 Likewise, the collection ID filtering module 212 corresponds to the relational  
12 expression 250C, and the version filtering module 214 corresponds to the relational  
13 expression 250D.

14 Referring to Figure 20, after the package filtering criteria are received, the  
15 method continues by applying 232 the package filtering criteria to an initial set of  
16 packages. In one embodiment, the initial set may include all of the packages  
17 querying the database 52 or any user-defined subset of the packages. In one  
18 embodiment, the user may initiate the filtering process by selecting an "OK" button  
19 152 (of Figure 22) or the like, which will cause the filtering modules having defined  
20 relational expressions 250 to apply the filtering criteria to the initial set of packages.

21 Thereafter, the method continues by displaying 234 a filtered list of packages  
22 to the user. Figure 23 illustrates the filtered list of selectable packages generated in  
23 response to the expression 250A of Figure 22. As provided in the filtering criteria,  
24 each of the listed packages is owned by "ADMF001," "USER001," or "SYSADM."

25 After the filtered list is displayed, the method continues by receiving 236 a  
26 user's selection of one or more of the packages. In one embodiment, the user may

1 select the desired packages by highlighting individual packages with a mouse or  
2 other pointing device. After the one or more packages are highlighted, the user may  
3 confirm his or her selections by means of an "add" button 254 or other suitable  
4 control, which will preferably cause the selected packages (of Figure 23) to be added  
5 to the selected package list 136, as shown in Figure 4. These selected packages are  
6 then used to define the initial set of query statements as previously described.

7 The process of selecting plans is substantially similar to the above-described  
8 process of selecting packages. In step 238, the method continues by receiving 238  
9 user-specified plan filtering criteria to be used in filtering an initial set of plans.  
10 Figure 25 illustrates an interactive display generated by the statement selection  
11 module 100 of Figure 19, and more particularly, by the plan selection module 202,  
12 before any plans have been selected. As previously described, the statement  
13 selection module 100 provides in one embodiment an "add" button 134, which the  
14 user may select to add one or more plans to the selected plan list 142.

15 As illustrated in Figure 26, when the "add" button 134 (of Figure 25) is  
16 selected, a set of customizable plan filtering criteria is preferably displayed. In one  
17 embodiment, the plan filtering criteria includes a number of relational expressions  
18 260 that are applied the list of plans in the database 52. Preferably, a relational  
19 expression 260 is provided for each filtering module, including the plan owner  
20 filtering module 218, the plan name filtering module 220, and the DBRM name  
21 filtering module 222.

22 For example, in the depicted embodiment, the plan owner filtering module  
23 218 corresponds to the relational expression 260A. Preferably, the expression 260A  
24 has the form:

25 "Plan owner:" OPERATOR VALUE  
26

1 where the OPERATOR field is selectable from a number of relational operators,  
2 such as "equal to," "not equal to," "greater than," "less than," "in," or "like," and  
3 the VALUE field stores the identifiers of one or more plan owners. In one  
4 embodiment, the OPERATOR field may be selected using a pull down menu or the  
5 like, while the VALUE field may be supplied using a text entry box or other suitable  
6 mechanism.

7 The completed expression 260A, i.e. "Plan owner: = ADMF001," is  
8 preferably interpreted by the plan owner filtering module 218 as an instruction to  
9 provide to the plan listing module 224 only those plans owned by "ADMF001."

10 Similarly, a user may further refine the selected list of plans by selecting  
11 filters corresponding to plan names (expression 260B) and DBRM names (expression  
12 260C). In the depicted embodiment, the plan name filtering module 220  
13 corresponds to the relational expression 260B and operates in substantially the same  
14 way as the plan owner filtering module 218. In other words, the OPERATOR field  
15 is selectable as described above, and the VALUE field stores the names of one or  
16 more plans. Likewise, the DBRM name filtering module 222 corresponds to the  
17 relational expression 260C.

18 For example, the completed expression 260C, i.e. "DBRM name: LIKE  
19 PROG%," is interpreted by the DBRM name filtering module 222 as an instruction  
20 to provide to the plan listing module 224 only those plans having a DBRM name  
21 such as PROG1, PROG2, PROG3, etc. The LIKE operator permits wildcards, such  
22 as "%," which can be used to represent any number of alphanumeric characters.

23 After the plan filtering criteria are received, the method of Figure 20  
24 continues by applying 240 the plan filtering criteria to an initial set of plans as  
25 described above. In one embodiment, the initial set may include all of the plans that  
26 query the database 52 or any user-defined subset of the plans. In one embodiment,

What is claimed and desired to be secured by United States Letters Patent is: